REMARKS

Claims 506-557, 559-607,609-615 are currently pending and have been rejected by the Examiner. Applicants traverse.

Claim Rejections under 35 USC § 102

The Examiner has rejected claims 506-517, 519-546, 548-557, 559-570, 572-594, 595-607, 609-613 under 35 U.S.C. 102(e) as being anticipated by Tremblay, (US 6,125,439).

Claim 506 includes the following limitations:

A method for processing instructions in a central processing unit (CPU) capable of executing instructions of a plurality of instruction sets, including a stack-based and a register-based instruction set, the method, comprising:

maintaining data for register-based instructions from the register-based instruction set and an operand stack for operands associated with stack-based instructions from the stack-based instruction set in a first register file, wherein at least some of the operands are moved between the register file and memory via at least one of an overflow and underflow mechanism;

maintaining an indication of a depth of the operand stack; and

executing the stack-based instructions and register-based instructions in an execution unit, including generating an exception in respect of selected stack-based instructions.

Regarding claim 506, the Examiner has stated that:

***As per** claim **506,** Tremblay (US 6,125,439) discloses:*
- a method for processing instructions in a central processing unit capable of executing instructions of a plurality of instruction set (Abstract), the processor is the central processing unit is shown in (col 5, lines 43-45)
- including a stack-based instruction (Abstract, lines 1-3) where "hardware processor" is the "stack-based instruction" (col 3, lines 53-56) and a register-based
instruction (col 17, lines 32-39), where "stack cache" is register file
(COI 17, lines 11-16) and the instructions are in stack-cache is the register-based
instruction
- maintaining data for register-based instruction from the register-based

instruction set (col3, line 64), where "operations on data are performed through the stack cache", inherently including maintaining data for register-based instruction as claimed
operand stack for a operands associated with stack-based instructions . . . overflow and underflow mechanism (col 18, lines 29-57, col 17, lines 10-15)
- maintaining an indication of a depth of the operand stack (col 9, lines 20-25), where "an instruction ***push this pointer onto operand stack"*** *inherently including maintaining an indication of a depth of the operand stack as claimed*
- executing the stack-based instructions and register-based instruction in an execution unit (abstract)
- including generating an exception in respect of selected stack-based instruction (col 3, lines 39-42).

In executing a new method, a hardware processor loads the execution environment on a stack in the background and indicates what portion of the execution environment has been loaded so far, e.g., simple one bit scoreboarding. Thus, the hardware processor tracks the information in the execution environment loaded on the stack. The hardware processor tries to execute the bytecodes of the called method as soon as possible, even though the stack is not completely loaded. If accesses are made to variables already loaded, overlapping of execution with loading of the stack is achieved. Thus, execution and loading continue until information in the execution environment needed for the execution is not on said stack as indicated by said tracking.

(Abstract,Tremblay)

Applicants respectfully submit that the abstract of Tremblay provides no support for the Examiner's conclusion that Tremblay teaches a hardware processor that is capable of executing instructions of a plurality of instructions sets. In this regard, Applicant's respectfully point out that while the hardware processor of Tremblay is capable of executing multiple instructions, each of said multiple instructions are from a single stack-based instruction set.

Further, the Applicants respectfully submit that the Examiner's analysis regarding register-based instructions does not bear scrutiny. In this regard, the Examiner is respectfully requested to consider the title of Tremblay, viz.,"PROCESS OF EXECUTING A METHOD ON A STACK-BASED PROCESSOR". The title clearly restricts applicability of the techniques of Tremblay to a "stack-based processor". Moreover, in the summary of the invention section (col. 4 lines 60-65) Tremblay states that *"the hardware processor directly implements a stack that supports the JAVA virtual machine* *stack-based architecture*" (emphasis added). Nowhere does Tremblay state that the hardware processor is capable of executing instructions of a register-based instruction set. In col 17, lines 11-16 Tremblay states that *"Stack cache **155** is a sixty-four entry thirty-two-bit wide array of registers that is physically implemented as a register file in one embodiment"*. However, the above statement does not support the Examiner's conclusion that the hardware processor of Tremblay is capable of executing instructions of a register-based instruction set. Registers are hardware storage elements. For example consider Tremblay col. 8, lines 15-20 where it is stated that *"depending on the specific implementation of the invention, the pointer may be implemented using a hardware register, a hardware counter, a software counter, a software pointer, or other equivalent embodiments known to those of skill in the art"*. This statement is consistent with a register being a hardware storage element. Thus, in col.17, lines 11-16, Tremblay is describing that the stack cache 155 may be physically implemented using an array of registers in a register file. However, this does not imply that because the stack cache 155 is implemented using registers, the hardware processor of Tremblay is capable of executing register-based instructions.

In col. 3, line 64, Tremblay states that *"operations on data are performed through the stack cache"*. Since, as argued above, the hardware processor of Tremblay only executes stack-based instructions, it follows that the data in the stack cache is the data for the stack-based instructions. The Examiner's conclusion that the above-quoted statement from col.3, line 64 provides inherent support for *"maintaining data for register-based instruction…"*, as recited in claim 506 is, with respect, not justified.

In respect of the limitation of *"executing the stack-based instructions and the register-based instructions…"* of claim 506, the Examiner merely refers to the abstract of Tremblay in parenthesis, without actually arguing why the Examiner believes that the abstract of Tremblay supports said limitation of claim 506. Applicants respectfully submit that nothing in the abstract of Tremblay supports the Examiner's conclusion in this regard. Moreover, as argued above, the hardware processor of Tremblay executes only stack-based instructions and not register-based instructions.

With regard to the limitation of generating an exception in respect of selected stack-based instructions, as recited in claim 506, the Examiner refers to Tremblay col. 3 lines 39-42, without any argument. Tremblay col. 3 lines 39-42 states that *"if desired, extra stages for memory access or exception resolution are provided"*. From this statement, Tremblay merely contemplates that the pipeline of the hardware accelerator may be extended to include additional stages to resolve exceptions. However, bearing in mind that the claim limitation in question recites generating an exception in respect of <u>selected</u> instructions, Tremblay fails to teach or suggest <u>selecting</u> any instruction in respect of which an exception is to be generated.

Based on the foregoing, Applicants respectfully submit that Tremblay does not teach all limitations of claim 506, and thus cannot anticipate claim 506.

Given that claims 507-529 depend on claim 506, it is respectfully submitted that these claims are also not anticipated by Tremblay.

Regarding independent claim 530, this claim includes the following limitations:

> *A method for processing instructions in a central processing unit (CPU), the method comprising:*
> *decoding instructions of a <u>first instruction set</u>;*
> *maintaining an operand stack associated with the instructions of the first instruction set in a register file including moving at least some operands between the register file and memory via at least one of an overflow and underflow mechanism;*
> *decoding instructions of a <u>second instruction set</u>;*
> *maintaining data associated with the instructions of the second instruction set in the register file;*
> *sending an output of the decoding of the instructions of the first and second instruction set, to an execution unit; and*
> *processing the output in the execution unit, including generating exceptions in respect of selected instructions of the first instruction set and processing the selected instructions for which exceptions were generated using a virtual machine.*
>
> (claim 506, emphasis added)

The Examiner refers to claim 530 on page 5 of the Office Action under rely. In particular, the Examiner states:

> ***For claim** 530,*
> *- decoding instruction . . . set (col 3, lines 20-30)*

- maintaining an operand stack ... underflow mechanism (col 3, line 64, col 18, lines 29-57, col 17, lines 10-1 5)
- decoding instruction . .. set (col 3, lines 20-30, col 3, lines 40-43)
- maintaining data . . . register file (col3, lines 63-65)
- sending an output . . . unit (col 18, lines 7-12)
- processing the output in the execution unit ... virtual machine (col 18 lines 7-12, col 3 lines 40-42, col 3 lines 60-65)

Applicants respectfully point out that in terms of 37 CFR 1.104, in rejecting a claim, the Examiner has the burden of explaining the pertinence of each reference. Applicants do not understand the pertinence of the portions of Tremblay referred to by the Examiner. Accordingly, the Examiner is respectfully requested to explain the pertinence of said portions of Tremblay. At any event, as argued above, the hardware processor of Tremblay executes only instructions of a single instruction set viz, instructions of a stack-based instruction set. Thus, the above-emphasized limitations of claim 530 are not found in Tremblay. Accordingly, it is respectfully submitted that Tremblay does not teach all limitations of claim 530. Further, given that claims 531-539 depend on claim 530, it is respectfully submitted that these claims are also not anticipated by claim 530.

Regarding claim 540, this claim includes that following limitations:

A method, comprising:
switching a processing system to an accelerator mode, wherein stack-based instructions are executed directly in hardware;
generating an exception in respect of a selected stack-based instruction while in the accelerator mode;

switching the processing system to a first native mode in which the selected stack-based instruction is processed within a virtual machine; and

switching the processing system to a second native mode upon a further exception, wherein in the second native mode the virtual machine is non-operative.

(claim 540, emphasis added)

In rejecting claim 540 the Examiner states the following:

For claim 540,
.switching a processing system to an accelerator (col 22, lines 18-35), wherein stack-based instructions are executed directly in hardware (col 5, lines 36-46)
- generating an exception ... accelerator mode (col 22, lines 28-40, col 23, lines 56-64)
- switching the process ... virtual machine (col21, lines 56-64, col22 lines 12-18).

Applicants, once again respectfully point out to the Examiner that, in terms of 37 CFR 1.104, the pertinence of Tremblay, with respect to claim 540, must be explained. At any event, the Examiner's analysis is prima facie incomplete since it omits any reference to the above-emphasized switching limitation upon a further exception. Moreover, the portion of Tremblay referred to by the Examiner describes accelerating a lookup switch instruction. Thus, the above-emphasized limitation of claim 540 is not found in Tremblay. Accordingly, it is respectfully submitted that Tremblay does not teach all limitations of claim 540. Further, given that claims 541-542 depend on claim 540, it is respectfully submitted that these claims are also not anticipated by claim 540.

Regarding claim 543, this claim includes the following limitations:

A method, comprising:
processing instructions of a plurality of instruction sets in a CPU having an execution unit and a register file, wherein at least one of the plurality of instruction sets being a stack-based instruction set and at least one of the

plurality of instruction sets being a <u>register-based instruction set</u>, using a <u>common program counter for the plurality of instruction sets</u>, the common program counter being stored in a common register; and
generating a branch taken signal to facilitate the processing of selected instructions of the stack-based and register-based instruction sets.

The Examiner's argument regarding claim 543 is reproduced below:

For claim 543
- processing instruction of a plurality of instruction sets in a CPU having an execution unit (Abstract, col 5 lines 36-48]) and a register file col 17, lines 11 - 15)
- wherein at least one of the plurality ... register-based instruction set (col 17, lines 32-39, col 17, lines 11-16)
- using a common program counter . . . common register (col 10,lines 5-1 0)

- generating a branch taken signal . . . sets (col 17, lines 38-24).

In response, the Applicants repeat the above arguments that the hardware processor of Tremblay processes instructions of a single instruction set viz., a stack-based instruction set. The hardware processor of Tremblay does not execute instructions of a register-based instruction set. Additionally, the limitation of a common program counter is not found in Tremblay as the hardware processor of Tremblay only executes instructions of a single instruction set.

Moreover, in col. 10, lines 5-10 Tremblay states

"Program counter register PC (FIG. 1) contains the address of the next instruction, e.g., opcode, to be executed. Locations on operand stack 423 (FIG. 4A) are used to store the operands of virtual machine instructions, providing both source and target storage locations for instruction execution."

With respect, it is not seen how the above quoted statement from Tremblay supports a <u>common</u> program counter. Based on the foregoing, it is respectfully submitted that Tremblay cannot antipate claim 543. Given, that claims 544-551 depend on claim 543, it is respectfully submitted that these claims are also not anticipated by Tremblay.

Regarding claim 551, this claim includes the following limitations:

In a processing system, comprising a central processing unit (CPU) having an execution unit and a register file capable of processing instructions of a plurality of instruction sets including a register-based instruction set and a stack-based instruction set, wherein an operand stack for the stack-based instruction set is maintained in the register file, operands are moved between the register file and memory due to at least one of an overflow and underflow mechanism, and wherein the processing system, further comprises a first state in which the CPU processes instructions using the register-based instruction set without a virtual machine, a second state in which the CPU processes using the non-stack-based instruction set within a virtual machine, and a third state in which the CPU processes instructions using the stack-based instruction set within the virtual machine, a method of operating the CPU comprising:

switching the processing system to the first state due to at least one of a reset command and a power-on condition;

switching the CPU to the second state;

processing instructions in the second state; and

upon encountering an exception while processing the instructions in the second state, switching the CPU to the first state.

The Examiner's arguments with respect to claim 551 are reproduced below:

***For claim 551,** (col 5, lines 35-45, col 5, lines 43-45, abstract, col 3, lines 53-56, col 17, lines 32-39, col 17, lines 11-16, col3 line 64, col21 lines 57-62, col6 lines 25-27). Tremblay discloses N-way switch system (col 21 lines 58-60), which shows the system can switch from one state to another sate, encountering exception (col23 lines 56-65).*

Tremblay describes a lookup switch accelerator 145 to process lookupswitch statements in the Java language (col. 22, lines 25-35). The lookup switch accelerator does not cause the hardware processor of Tremblay to switch from one state to another upon encountering an exception as argued by the Examiner. In this regard, the Applicants aver that even if the Examiner's argument that the lookup accelerator did switch the hardware accelerator of Tremblay upon encountering an exception were correct, this would be of no consequence as *"switching from one state to another state, encountering* *exception"* (sic) is not a claim limitation of claim 551. In the circumstances, the Examiner has failed to show that Tremblay teaches all limitations of claim 551. For this reason, it is respectfully submitted that claim 551 is not anticipated by Tremblay.

Given that claims 552-557 depend on claim 551, it is respectfully submitted that these claims are not anticipated by Tremblay. The remaining claims have limitations similar in scope to the above-discussed limitations. Accordingly, it is respectfully submitted that the remaining claims are also not anticipated by Tremblay.

Claim Rejections under 35 USC § 103

The Examiner has rejected claims 518,547,571,595,614, and 615 under 35 U.S.C. 103(a) as being unpatentable over Tremblay. In response, the Applicants repeat the above arguments that Tremblay does not teach or suggest all limitations of these claims.

It is respectfully submitted that in view of the remarks set forth herein, all rejections have been overcome. All pending claims are now in condition for allowance, which is earnestly solicited. Authorization is hereby given to charge our Deposit Account 503437 for any charges that may be due. Furthermore, if an extension is required, then Applicant hereby requests such an extension.

Respectfully submitted,

HAHN AND MOODLEY LLP

[illegible]

Vani Moodley (Registration No. 56,631)

Dated: 3/14/, 2005

Suite 180
800W El Camino Real
Mountain View,94040